

TONTTIEN MYYNTIPALVELU VERKKOYMPÄRISTÖSSÄ

LAHDEN AMMATTIKORKEAKOULU

Tekniikan ala

Mediatekniikka

Tekninen visualisointi

Opinnäytetyö

Kevät 2011

Kristian Lindén

lindkris@lpt.fi

Lahden ammattikorkeakoulu
Mediatekniikan koulutusohjelma

LINDÉN, KRISTIAN:

Tonttien myyntipalvelu verkkoympäristössä

Mediatekniikan opinnäytetyö, 36 sivua, 0 liitesivua

Kevät 2011

TIIVISTELMÄ

Opinnäytetyössä selvitetään mitä asioita pitää ottaa huomioon nykyaikaisen verkkopalvelun suunnittelussa. Mitkä tekniikat ja osa-alueet ovat tärkeimpiä kun palvelua aletaan suunnitella.

Ensin käydään läpi tarvittavia tekniikoita itse palvelun toteuttamiseen ja sisällön esittämiseen rikkaassa muodossa. Tämän jälkeen paneudutaan visuaaliseen suunnitteluun, suunnittelun teoriaan ja eri rakenteellisiin seikkoihin.

Lopuksi case-osuudessa tutustutaan tonttien myyntipalvelun konseptin suunnitteluun ja toteutukseen. Tarkastellaan palvelun lähtökohtia, mahdollisuuksia ja ilmenneitä ongelmia.

Palvelu rakennettiin yhteistyössä kahden muun opiskelijan kanssa. Lopputuloksena syntyi täysin toimiva palvelun pohja, joka esiteltiin myös Raksa-messuilla 2011 keväällä.

Avainsanat: verkkosuunnittelu, html, css, front-end, saavutettavuus

Lahti University of Applied Sciences
Degree Programme in Media Technology

LINDÉN, KRISTIAN: Lot selling service for the web environment

Bachelor's Thesis in Media technology

36 pages

Spring 2011

ABSTRACT

This Bachelor's thesis deals with how to design a modern web service. The objective was to examine what techniques and factors are the most important for the process.

Different techniques were explored, in order to produce the service and to present the content in a rich manner. Those techniques are markup languages, style sheets and more advanced programming languages for example scripting. After covering the techniques, the thesis discusses visual design, design theory and different structural aspects. These sections deal with for example web-typography, user interface design and color theory.

The last part is the case section, where designing and producing the concept for a lot selling service is covered. The starting point for the service is examined, as well as the potentials of the service and also the problems that emerged.

Keywords: web design, html, css, front-end, accessibility

SISÄLLYS

1	JOHDANTO	1
2	WEB-TEKNIIKAT VERKKOPALVELUN TOTEUTTAMISEEN	3
2.1	Web-tekniikoiden nykypäivä	3
2.1.1	HTML	3
2.1.2	CSS	3
2.1.3	JavaScript	4
2.1.4	DOM	5
2.1.5	Liitännäiset	5
2.2	Katsaus tulevaisuuteen	6
3	RIKKAAN SISÄLLÖN ESITTÄMINEN VERKKOPALVELUSSA	7
3.1	Flash	7
3.2	HTML5 – canvas	8
3.3	CSS3	9
3.4	Silverlight	12
3.5	JavaScript kirjastot – jQuery	12
3.6	Google Maps API	13
4	VERKKOSIVUN VISUAALISUUS JA RAKENNE	15
4.1	Käytettävyyden merkitys	15
4.2	Sivun suunnittelu	16
4.3	Käyttöliittymäsuunnittelu	18
4.4	Värit	19
4.4.1	Värien valinta	21
4.5	WWW-Typografia	22
4.5.1	Typografian saavutettavuus	22
4.5.2	Fontin valinta	24
5	CASE	26
5.1	Lähtökohdat ja tavoitteet	26
5.2	Työn kuvaus	27
5.2.1	Verkkopalvelun suunnittelu	27
5.2.2	Verkkopalvelun tekniikka	31
	YHTEENVETO	35
	LÄHTEET	36

1 JOHDANTO

Internetpalveluiden kehitys on ollut nopeaa viime vuosina ja projektien vaatimukset lähestyvät entistä enemmän työpöytäsovelluksia. Kun halutaan tarjota käyttäjälle ja mahdolliselle asiakkaalle mieleenpainuva sekä käyttäjäystävällinen kokemus, joudutaan perinteisen verkkosivun kuvausta laajentamaan kohti rikkaampaa sisältöä. Tarvitaan esimerkiksi innovatiivisempia tapoja esittää tieto käyttäjälle, kehittyneitä sivuston räätälöintimahdollisuuksia ylläpitäjälle, jotta palvelun sisältöä ja toiminnallisuuksia olisi mahdollisimman helppo muuttaa sekä muiden palveluiden sisällyttämistä sivustolle.

Tässä opinnäytetyössä tutustutaan rikasta sisältöä vaativan verkkopalvelun toteutukseen visuaalisuuden ja front-end kehityksen näkökulmasta. Tässä front-end kehityksellä tarkoitetaan selainpuolen ohjelmointia sekä kuvauskielten käyttöä. Kirjallisessa osuudessa pyritään selvittämään, mitkä nykyaikaisista tekniikoista ovat soveltuvimpia sellaisen palvelun toteuttamiseen ja mitkä käyttöliittymäsuunnittelun alueet nousevat tärkeimmiksi. Tarkoituksena ei ole selvittää kaikkia olemassa olevia tekniikoita tai mahdollisuuksia, vaan enemmänkin verrata keskenään tämän päivän relevantteja vaihtoehtoja. Vaikka sisällön suunnittelu on myös merkittävä osa itse palvelun suunnittelua, niin siihen ei paneuduta, koska sisällöntuotanto tulee tämän opinnäytteen ulkopuolelta.

Tasapaino tekniikan ja visuaalisten - jopa taiteellisten ansioiden välillä on yksi tärkeimpiä WWW-suunnittelun osia; kuinka asettaa sopivasti painoarvoa tekniselle helppokäyttöisyydelle viemättä kaikkea mielenkiintoa visuaalisesta ulosannista, ja toisinpäin. Myös tähän ajatusleikkiin pyritään löytämään vastauksia kirjallisessa osuudessa.

Lopuksi läpikäydään case-projekti, jossa kirjallisessa osuudessa opittuja asioita koitetaan käytännössä. Tarkoituksena on tuottaa visuaalinen suunnitelma, käyttöliittymäsuunnitelma, kuvitukset sekä toteuttaa palvelun front-end-ohjelmointi yhteistyössä toisen opiskelijan kanssa. Case-projektissa pyritään nimenomaan mahdollisimman visuaalisesti miellyttävään lopputulokseen, joka ei kuitenkaan astu

käytettävyyden varpaille sekä mahdollisesti jopa löytää uudenlaisia ratkaisuja esittää tietoa käyttäjälle.

2 WEB-TEKNIIKAT VERKKOPALVELUN TOTEUTTAMISEEN

2.1 Web-tekniikoiden nykypäivä

2.1.1 HTML

HTML, eli HyperText Markup Language, on WWW:n yhdistävä kieli. Sen yksinkertaisilla tageilla voidaan antaa pohja lähes minkälaiselle palvelulle tahansa. Pieni tai iso palvelu, tarvitaan HTML:ää aina. (Keith, 2010)

HTML5 on kielen uusi versio, johon keskitytään edempänä tarkemmin. HTML kieltä on uudistettu useaan kertaan eikä HTML 5 ole sen ensimmäisiä uudistuksia. Se on itse asiassa kehittynyt koko olemassaolonsa ajan. (Keith 2010.)

Kuten myös itse WorldWideWeb, HTML on Sir Tim Berners-Leen keksintö. Vuonna 1991 hän kirjoitti ensimmäisen säännösten HTML:ää varten. Idea ei sinänsä ollut kuitenkaan uusi vaan Berners-Lee lähinnä jatkokehitti SGML:n eli Standard Generalized Markup Languagea ideaa tageista. (Keith 2010.)

HTML on pohjaltaan tekstiä, jota voidaan tuottaa missä tahansa tekstieditorissa, vaikka sen tuottamiseen on olemassa myös varsinaisia HTML-editoreita. Rakenteensa HTML saa kulmasulkeilla erottamisesta ja avainsanojen käyttämisestä. Tämän yksinkertaisen pohjaidean takia HTML:ää voi käyttää lähes rajattomasti moneen eri tarkoitukseen. (Wikipedia 2011.)

2.1.2 CSS

CSS eli Cascading Style Sheets on web-ympäristöön tarkoitettu tyyliohjeiden laji. Tällä säännöstöllä määritetään yleensä verkkopalvelun ulkonäön peruspiirteet, kuten elementtien sijoittelu, väri, typografia ja niin edelleen. CSS on ensisijaisesti

suunniteltu erottamaan dokumentin tyylimäärittelyt sen sisällöstä. Tästä erottelusta kumpuaa CSS:n tärkein hyöty. Esimerkiksi web-sivujen ulkoasua voi muuttaa koskematta sisältöön, ja esimerkiksi kaikki värit voidaan vaihtaa yhdestä ja samasta paikasta. CSS:llä on mahdollista myös esittää sisältö monella eri tavalla, kuten Braille-pohjaisena tai ääni-pohjaisena näkörajoitteisille. Braille-pohjaisuudella tarkoitetaan sitä kun erillinen laite lukee sivuston tyylimäärittelyt ja kääntää sen sokeille tarkoitetulle pistekirjoituslaitteelle. Näin näkörajoitteinen voi lukea verkkopalvelun sisällön. Se on tärkeää, kun verkkopalvelussa tähdätään 100 %:n saavutettavuuteen. (Wikipedia 2011.)

CSS:stä on olemassa kaksi versiota: CSS1 ja CSS2. Tällä hetkellä ollaan kehittämässä CSS2:n päivitysversiota CSS2.1:stä sekä CSS3:sta. Näiden tyylikielten eri versioita tapaa verkossa jo nyt käytössä runsaasti, mutta kuten HTML:n kohdalla, kehitys on jatkuvaa. Tällä hetkellä kilpailevista selainmoottoreista Gecko jota Mozilla käyttävät, WebKit jota Safari sekä Googlen Chrome käyttää, Presto jota Opera käyttää sekä Internet Explorer kaikki tukevat kohtuullisesti CSS2.1:n suosituksia. CSS3:n tukemisesta ollaan kuitenkin vielä kaukana, johon palataan tarkemmin myöhemmin. (Wikipedia 2011.)

2.1.3 JavaScript

JavaScript on pääasiassa web-ympäristöön kehitetty komentosarjakieli, joka pohjautuu alun perin ECMAScript-kieleen. JavaScript on oliopohjainen kieli, jonka tärkein ominaisuus on mahdollisuus lisätä dynaamista toiminnallisuutta web-sivuille. Sen syntaksi muistuttaa c-kieltä, ja se lainaa nimeämiskäytäntöjä Javalta, mutta muuten JavaScript ei liity mitenkään muihin kieliin. (Wikipedia 2011.)

JavaScriptin käyttö web-sivuilla on lisääntynyt viime vuosina paljon. Sen ja ulkoisten kirjastojen kuten jQuery:n avulla kehittäjiä on helppo toteuttaa yleisimpiä dynaamisia toiminnallisuuksia sivustoille. Se on oikein käytettynä kevyttä ja sitä tuetaan laajasti. Myös mobiiliselaimet tukevat lähes poikkeuksetta JavaScriptiä, joten se on vartenotettava vaihtoehto erillisille liitännäisille. Mobiililaitteiden osuus kaikesta internet liikenteestä on koko ajan kasvussa, joten on ensiarvoisen

tärkeää että sisältö on saavutettavissa myös näillä laitteilla. Tämän myötä todennäköisesti myös JavaScriptin merkitys kasvaa edelleen. StatCounterin mukaan mobiiliselausten osuus on vasta alle 5 prosenttia kaikesta selauksesta, mutta suuntaus on kasvava.

2.1.4 DOM

DOM, eli Document Object Model on alusta- ja kieliriippumaton rajapinta joka mahdollistaa HTML-dokumenttien muokkauksen. Yhdistettynä esimerkiksi JavaScriptiin sen avulla voidaan toteuttaa dynaamista toiminnallisuutta ja vuorovaikutteisia web-sivuja, jotka eivät vaadi jatkuvaa palvelinyhteyttä. (Wikipedia 2011.)

Web-selaimet eivät ole velvollisia käyttämään DOM:ia HTML-dokumenttien esittämiseen. DOM:in avulla JavaScript kuitenkin ymmärtää HTML-sivun ja sen tilan, joten se on vaatimus jos web-dokumentin sisältöä halutaan muokata dynaamisesti. Tämän seurauksena DOM kuuluu nykyaikaisten web-selainten vaatimukseen. (Wikipedia 2011.)

2.1.5 Liitännäiset

Liitännäinen on ohjelmistokomponentti, joka lisää määrättyjä mahdollisuuksia laajempaan ohjelmistokokonaisuuteen, kuten esimerkiksi tässä tapauksessa web-selaimeen. Liitännäisten avulla voidaan helposti tuoda saataville sisältöä, joka ei pelkästään selaimen ominaisuuksin onnistu. Liitännäisten käytöllä voidaan myös pienentää itse sivuston kokoa sekä pitää erillään rikkaan sisällön vaatima koodi. Täten liitännäiset ovat tärkeä osa rikkaan sisällön esittämistä verkossa. On kuitenkin muistettava että liitännäiset vaativat aina erillisen asennuksen, mutta aina tämä ei ole edes mahdollista, kuten tietyillä mobiilialustoilla tai tietokoneilla joilla käyttäjällä ei ole asennusoikeuksia. Yleisimpiä liitännäisiä rikkaan sisällön esittämiseen ovat Adobe Flash, Microsoft Silverlight ja Java appletit. (Wikipedia 2011.)

2.2 Katsaus tulevaisuuteen

Web-tekniikoiden tulevaisuus on mielenkiintoinen mutta epävarma. Kukaan ei osaa ennustaa, mitkä tekniikat häviävät ja mitkä säilyvät sekä parantavat asemaansa. Yleisesti kuitenkin ajatellaan että liitännäisten osuus tulee vähenemään ja suoraan web-selaimessa toimivat tekniikat yleistymään. Siitä on jo nyt viitteitä isojen ohjelmistoyritysten siirtyessä pikkuhiljaa pois ylimääräisistä liitännäisistä, esimerkiksi Youtube- ja Vimeo-videopalveluiden muuttuminen asteittain HTML5-videoihin.

Kun pohditaan sitä, mikä tekniikka tulee olemaan tulevaisuudessa tärkein, joudutaan asiaa katsomaan monesta suunnasta. Kuten Brightcove videoalustan toimitusjohtaja Jeremy Allaire erittelee artikkelissaan ”The Future of Web Content – HTML 5, Flash & Mobile Apps”, on kyse ennemminkin siitä, mikä tekniikka ottaa haltuunsa minkäkin osa-alueen. Hänen mielestään Flashiä ei esimerkiksi tulla korvaamaan tietyillä osa-alueilla. HTML5-pohjaiset ratkaisut toimivat paremmin ns. Web Productivity Appseissä, kun taas Flashin yliveraisuus näkyy esimerkiksi rikkaissa tiedon visualisoinneissa. (Allaire 2010.)

On totta että ilman liitännäisiä toimivat ratkaisut ovat osoittaneet joissain soveluksissa yliveraisuutensa, mutta kuluu vielä vuosia, ennen kun ne pääsevät esimerkiksi Flashin monipuolisuuteen. Allaire luettelee myös, että ison teollisuuden on vielä todella vaikea siirtyä pois Flashin kaltaisista ympäristöistä, ja ainoa selkeä iso alue on mobiilialustat, joissa Flashista siirrytään pois. Niissä osin Applen painostuksen vuoksi on jo päädytty tilanteeseen, jossa liitännäisiä ei suositella, mikä on otettava huomioon tulevaisuuden palveluita suunniteltaessa. Allairen mukaan kriittinen massa esimerkiksi HTML5: n hyväksi nähdäänkin ensiksi juuri mobiilialustoissa ja niiden webselaimissa. (Allaire 2010.)

3 RIKKAAN SISÄLLÖN ESITTÄMINEN VERKKOPALVELUSSA

3.1 Flash

Adobe Flash on multimedia-alusta, jolla voidaan lisätä animaatiota, ääntä, videota ja vuorovaikutteisuutta web-sivustoille. Flashia käytetään paljon esimerkiksi mainontaan sekä peleihin mutta myös työkaluna rikkaiden internet sovellusten tuottamiseen. (Wikipedia 2011.)

Flash käsittelee vektori- sekä rasterigrafiikkaa, ja se pystyy esittämään niiden avulla animaatiota, tekstiä, piirroksia ja kuvia. Se tukee myös sekä videon että äänen esittämistä, niin paikallisesti kuin suoratoistettunakin. Käyttäjän syöttämää tietoa se pystyy tallentamaan hiiren, näppäimistön, mikrofonin tai kameran kautta. Kaikkea näitä pystytään käsittelemään Flashin sisältämällä oliopohjaisella ActionScript-ohjelmointikielellä. (Wikipedia 2011.)

Flash on siis hyvin monipuolinen tekniikka, ja sen ominaisuuksien ansiosta sillä voidaan toteuttaa lähes mitä vain. Vapaiden käsien antaminen suunnittelijoille on kuitenkin joidenkin mielestä yksi Flashin suurimpia heikkouksia. Käytettävyyss-konsultti Jakob Nielsen kirjoitti jo vuonna 2000 artikkelissaan ”Flash: 99 % Bad” kuinka liika vapaus johtaa yleensä suunnittelustandardien laiminlyömiseen. Iso osa Nielsenin luettelemista heikkouksista on myöhemmin hävinnyt, mutta perustavaa laatua oleva ongelma on olemassa edelleen. Flashin vapaus suorastaan hokuttelee suunnittelijoita suunnittelemaan käyttöliittymiä jotka eivät ole erityisen käytettäviä suuren yleisön kannalta. (Nielsen 2000.)

Edellä mainitut suunnitteluongelmat ovat tietenkin ohitettavissa hyvällä suunnittelulla. Huono suunnittelu ei sinänsä ole alustan ongelma. Flashia on kuitenkin kritisoitu entistä enemmän myös muilla osa-alueilla. Applen toimitusjohtaja Steve Jobs luettelee avoimessa kirjeessään useita syitä Flashin hylkäämiseen heidän mobiililaitteistaan. Teknisestä näkökulmasta suurimmiksi kompastuskiviksi nousee

kykenemättömyys rautapohjaiseen kiihdytykseen. Mobiililaitteissa on tärkeää, että video pystytään purkamaan laitteessa itsessään eikä ohjelmistossa. Jos video puretaan vasta ohjelmistossa, vähenee akun varauksen kesto dramaattisesti. Tämän lisäksi Flash on alun perin suunniteltu tietokoneille hiiripohjaiseen käyttöön ja siksi tuottaa ongelmia mobiililaitteiden kosketuspohjaisissa käyttöliittymissä. Kosketuspohjaiset laitteet eivät esimerkiksi tunne erilaisia kursorin kohdistamiskäytäntöjä, johon taas monet flash-pohjaiset käyttöliittymät perustuvat. Flashin lähtökohtiin kuuluva kyky esittää sisältöä videon päällä on myös yksi suurimmista suorituskyvyn ongelmista, joka on toiminut lähes yksinomaan syynä isojen video-palveluiden siirtymiselle HTML5 canvas videoihin. (Jobs 2010.)

3.2 HTML5 – canvas

Canvas on HTML5:n myötä esitelty uusi elementti. Sen HTML-osuus on hyvin yksinkertainen, ja siinä määritellään ainoastaan tunniste, koko ja mahdollinen vaihtoehtoinen sisältö selaimille, jotka eivät osaa canvasta näyttää. Kaikki varsinainen toiminta tapahtuu JavaScriptin avulla. (Keith 2010, 23 – 29.)

Kun selaimiin esiteltiin kuvien lisäysmahdollisuus, se muutti kertaheitolla verkkokokemuksen. Kuviksi voi lisätä myös gif-animaatioita, joiden tyyliä voi muuttaa JavaScriptillä tai ne voi luoda dynaamisesti palvelimella, mutta kun ne esitetään selaimessa, niitä ei voi enää päivittää. Tähän pulmaan HTML5 canvas on nimenomaan suunniteltu ratkaisuksi. (Keith 2010, 23-29.)

JavaScriptissä viitataan canvas-elementtiin ja sen sisältöön. Sisällöllä tarkoitetaan tässä kohtaa API:a eli Application Programming Interfacea. Canvasin 2D API:lla on mahdollista piirtää kaikenlaisia kuvioita hyvinkin monipuolisesti. Se sisältää vektoripiirto-ohjelmista tuttuja työkaluja kuten: viivoja, täyttövärejä, liukuvärejä, varjoja, muotoja sekä Bezier-kurveja. Ero löytyy siitä, että graafisen käyttöliittymän sijasta kaikki määritellään JavaScriptillä. (Keith 2010, 23-29.)

Teoriassa canvas-elementillä voisi piirtää kaiken saman, kun esimerkiksi Adoben Illustrator-ohjelmalla. Se olisi kuitenkin äärimmäisen työlästä, eikä niinkään can-



























vaksen ydinidea. Canvaksen varsinainen hyöty tulee sen päivitettävyydestä. Se pystyy piirtämään uutta sisältöä käyttäjän valintojen mukaisesti. Koska sen on mahdollista vastata käyttäjän tekoihin, sillä voi toteuttaa erilaisia työkaluja, pelejä ja rikasta sisältöä, jotka ennen vaativat jonkin liitännäisen, kuten Flashin, käyttöä. Canvas elementtiä tukevat lähes kaikki modernit selaimet, paitsi Internet Explorer 8, joten sen käyttäminen ainoana tiedonesitystapana on vielä hieman kyseenalaista. Internet Explorerille on mahdollista saada tuki canvasille kiertotietä, mutta viimeistään versiossa 9 pitäisi olla natiivituki. (Keith, 2010, 23-29; Wikipedia, 2011)

3.3 CSS3

CSS3 on CSS:n uusin versio. CSS3 ei kuitenkaan ole missään nimessä valmis standardi vaan W3C:n antama joukko suosituksia. CSS3:n yksi suurimmista ideoista onkin se, että sitä ei kehitetä kokonaisena, vaan se on jaettu pienempiin erillisiin moduuleihin, joita selainvalmistajien on helpompi omaksua omaan tahtiin. Suunnittelijoille tästä on suurta hyötyä, kun he voivat kokeilla eri ominaisuuksia nopeammin. Ei ole tarvetta odottaa, että moduuleista tulee varsinaisia virallisia suosituksia. (Cederholm 2007, 5-6.)

Vaikka CSS3 ei ole vielä kokonaan valmis suositus ja sen selaintuki on vähintäänkin hajanaista, sitä voi kuitenkin ihan hyvin käyttää, kunhan vaan tietää missä tilanteissa ja miten. Cederholm erottelee sivuston osa-alueet suunnittelun näkökulmasta kahteen kategoriaan: kriittiseen ja ei-kriittiseen. Kriittisiä osuuksia ovat esimerkiksi saavutettavuus, käytettävyys ja sivuston brändi. Näihin ei olekaan syytä suoraan sekoittaa CSS3:sta, koska sen toimivuutta kaikilla käyttäjillä ei voida taata. Sen sijaan kokemuskerroksella sijaitsevat ei-kriittiset alueet, kuten interaktio, visuaaliset palkinnot, palaute sekä animaatio, ovat oivia kohteita CSS3:lle. Ennen jos haluttiin luoda kevyttä animaatiota tai visuaalisia palkintoja käyttäjälle, jouduttiin turvautumaan semanttisen HTML:n sekä CSS:n lisäksi esimerkiksi Flashiin. CSS3:lla on mahdollista esimerkiksi toteuttaa pieniä animaatioita, tyyllitelyjä kuten varjoja ja vaikka kerrostettuja taustakuvia. (Cederholm 2010, 7-8.)

Tärkeintä on siis erottaa tilanteet, joissa CSS3:sta kannattaa käyttää. Jos CSS3:än syntaksi kirjoitetaan oikein, sen käyttö ei myöskään aiheuta ongelmia vanhempien selainten kanssa. Selaimet, jotka eivät sitä ymmärrä, sivuuttavat sen vain täysin ja käyttävät seuraavia määrittelyjä tyyleille. Kunhan muistaa, että sitä ei kannata käyttää mihinkään kriittiseen toimintoon, voi CSS3:sta hyvin käyttää jo nyt rikkaitaan käyttäjäkokemusta sivustoilla. (Cederholm 2010, 35-36.)

Ominaisuus	Selaintuki				
Border-radius	 3+	 1+	 9	 10.5+	 3+
Text-shadow	 2+	 3.1+	 	 9.5+	 1.1+
Box-shadow	 3+	 3.5+	 9	 10.5+	 3+
Multiple background images	 2+	 3.6+	 9	 10.5+	 1.3+
Opacity	 1+	 1.5+	 9	 9+	 1.2+
RGBA	 3+	 3+	 9	 10+	 3.2+

Kuvio 1 CSS3:n yleisimmät ominaisuudet ja niiden selaintuki (Lindén 2011.)

3.4 Silverlight

Microsoft Silverlight on sovelluskehys rikkaan sisällön tuottamiseen ja esittämiseen verkossa. Sen pääpaino on multimediasa, animaatioissa sekä grafiikassa, ja se muistuttaa tarkoitukseltaan Flashia. Flashin tavoin Silverlight sisällön katsomiseen tarvitsee erikseen ladattavan liitännäisen, joka löytyy kaikille merkittävillä selaimilla, Windows-, OS X- sekä Linux-ympäristöön. Alun perin Silverlight kehitettiin median suoratoistoa varten mutta versiosta kaksi eteenpäin Silverlightiin on lisätty paljon vuorovaikutteisuutta sekä tuki .NET-ohjelmointikielelle. Tämän hetken uusin versio neljä julkaistiin huhtikuussa 2010. (Wikipedia 2011.)

Teknisesti Silverlight on pätevä vaihtoehto esimerkiksi sen suoranaismmalle kilpailijalle Flashille. Sen WPF-animaatiomalli perustuu aikaan Flashin kehysmallin sijaan. Tämä keventää animaation esittämistä huomattavasti koska animaation alku- ja loppupisteen välille ei tarvitse laskea jokaista kuvaa erikseen. Kehysmallin suurimpia heikkouksia on myös sen ajoituksen riippuvuus tietokoneen laskentatehosta, jolla esitystä katsotaan. WPF:n lisäksi Silverlight sisältää hyviä ominaisuuksia, kuten lokalisaatiomahdollisuus, hyvä video- ja äänikoodekki tuki, helppo käyttöönotto sekä rautapohjainen kiihdytys hi-definition videolle. (Mary 2010.)

Silverlightin tekninen yliveraisuus on suhteellisen selkeää vertaillen ominaisuuksia. Sen ainoa suuri heikkous on hieman vaatimaton levinneisyys. Kuten Statowl-verkkopalvelun tilasto Joulukuulta 2010 osoittaa, on Silverlightin levinneisyys vain noin 60 %, kun esimerkiksi Flashin levinneisyys on hyvin lähellä 100 %:a. (StatOwl 2011.)

3.5 JavaScript kirjastot – jQuery

jQuery on selainrajoittamaton JavaScript-kirjasto joka on suunniteltu yksinkertaistamaan HTML-sivustojen skriptausta. Se on tämän hetken käytetyin JavaScript kirjasto, ja se on käytössä yli 41 %:ssa 10 000 suosituimman sivuston joukossa. JQueryn syntaksi on suunniteltu tekemään dokumentissa navigointi helpoksi, valitsemaan DOM-elementtejä, luomaan animaatioita, käsittelemään tapahtumia

ja kehittämään Ajax-sovelluksia. Näiden lisäksi jQuery antaa mahdollisuuden luoda liitännäisiä ydintoiminnan päälle ja siten edelleen lisäämään toiminnallisuuksia. Kaiken tämän kanssa jQuery tarjoaa mahdollisuuden niin pienimuotoisen animaation tekemiseen kuin korkeatasoisten teemoitettavien widgettien luomiseenkin. JQueryn vahvuus on sen kyky auttaa monipuolisesti sivuston toteutuksen kanssa. Sillä voi automatisoida yleisiä toimintoja ja yksinkertaistaa monimutkaisempia. (Chaffer & Swedberg 2009, 7-10; Wikipedia 2011.)

Vaikka jQuery onkin äärimmäisen monipuolinen, siitä löytyy kuitenkin tarvittavaa johdonmukaisuutta, ja suurin osa sen rakenteesta on lainattu HTML:stä sekä CSS:stä. JQueryn kehityksessä on myös käytetty monia eri strategioita, joilla koodia on saatu yksinkertaistettua, pienennettyä ja tehostettua. JQueryn toimintaan vaadittava paketti onkin pieni, ainoastaan alle 20 kilobittiä pakattuna. Jos halutaan käyttää ulkoisten kehittäjien liitännäisiä, koko tietenkin kasvaa, mutta pysyy yleensä silti hyväksyttävissä rajoissa. (Chaffer & Swedberg 2009, 7-10; Wikipedia 2011.)

3.6 Google Maps API

Kun Google Maps API julkaistiin kesäkuussa 2005 ja se antoi kehittäjille mahdollisuuden lisätä Google Mapsin toiminnallisuutta sivuillensa. Google Mapsin käyttö sivuilla on ilmaista, mutta Googlen tiedotteen mukaan sillä on mahdollisuus lisätä mainoksia palveluun tulevaisuudessa. Google Mapsin API suunniteltiin alun perin vain JavaScriptille mutta viime aikoina se on laajentunut kattamaan myös Flash API:n Flash sovelluksia varten. Google Maps API on eniten käytetty verkkosovelluskehitys API, ja sitä käytetäänkin yli 350 000 sivustolla. (Wikipedia 2011.)

Google Mapsin avulla on mahdollista saada toimiva paikannusjärjestelmän sisältävä karttapalvelu omalle verkkosivulle. Tämän lisäksi sen avulla on mahdollista saada geokoodeja, muodostaa ajo-ohjeita tai tarkastella korkeuseroja. Sen API:n avulla on mahdollista muokata sen tiedonesitystä ja esimerkiksi lisätä erilaista materiaalia itse kartan päälle. Tämä avaa moninaisia mahdollisuuksia verkkosivu-

4 VERKKOSIVUN VISUAALISUUS JA RAKENNE

Aluksi on syytä tarkentaa internet-sivun ja –sivuston eroavaisuus. Jakob Nielsen kertoo Tim Berners-Leen alkuperäisestä WWW-suunnitelmasta, että kaikki informaatio esitetään käyttäjälle sivun avulla ja että sivu on selailun perusyksikkö. Sivu siis tarkoittaa esimerkiksi kotisivua, kun taas sivusto on eri alisivujen yhtenäinen rakenne. (Nielsen 2000, 85.)

Sivun suunnittelua voi lähestyä kahdesta eri lähtökohdasta, enemmän taiteellisesta tai sitten teknisestä suunnasta. WWW-sivun suunnittelijan täytyykin jatkuvasti tasapainoilla näiden kahden välillä. Sivu tarvitsee myös taiteellisia ja visuaalisesti mielenkiintoisia osia, mutta lähtökohtaisesti on kyse tekniikasta. Jakob Nielsen painottaakin helppoutta tekniikan näkökulmasta, enemmän kuin taiteellisia ansioita. (Nielsen 2000, 11.)

Tässä opinnäytteessä keskitymme nimenomaan sivun suunnitteluun sivuston sijasta, koska toteutettavan projektin rakenne muistuttaa enemmän pelkästään yhtä sivua kuin perinteistä sivustoa. Yleensä sivuston suunnittelu on tärkeämpää kuin yksittäisen sivun elementtien sijoittelu, koska käyttäjä ei löydä haluamalleen sivulle ylipäättänsä, jos suunnittelu on suoritettu hutiloiden. Tämän projektin erikoislaatuisuuden vuoksi sitä ei kuitenkaan käsitellä.

4.1 Käytettävyyden merkitys

Webissä tärkeintä on käytettävyys. Tämä tarkoittaa yksinkertaisesti sitä, että jos käyttäjä ei löydä jotakin tuotetta, hän ei myöskään osta sitä. (Nielsen 2000.)

Kun tiedon ja sivustojen määrä jatkaa kasvuaan räjähdysmäisesti, nousee sivujen käytettävyys entistä tärkeämmäksi asiaksi. Perinteisessä fyysisessä kaupankäynnissä tuotteen ostaja saattoi arvioida käytettävyyden vasta kotona avattuaan tuot-

teen pakkauksen. Esimerkiksi uuden television ostaja huomaa ehkä vasta kotona kuinka huono kaukosäätimen käyttöliittymä todellisuudessa on. (Nielsen 2000, 10.)

Internetissä tilanne on kuitenkin täysin päinvastainen. Mahdollinen ostaja joutuu heti ensimmäiseksi käyttöliittymän armoille, ennen kun edes ajatellaan ostopäätöstä. Kun tarjonta on valtavaa ja kasvaa koko ajan, miksi käyttäjä viettäisi yhtään ylimääräistä aikaa sivustolla joka turhauttaa häntä. On siis varmasti sangen selkeää kuinka tärkeää helppo käytettävyyys on Internetiin suunniteltaessa. (Nielsen 2000, 10-11.)

Käytettävyyttä voidaan tutkia helposti esimerkiksi järjestämällä yksinkertaisia käyttäjätestauksia. Yksinkertaisimmillaan kyse on siitä, onko informaation löytäminen helpompaa vaihtoehdon A vai B avulla. Näin saadaan selville esimerkiksi onko käyttäjällä vaikeuksia löytää tarvitsemansa tieto tai toimittaa jokin haluamansa toiminto. Tämä ei kuitenkaan niinkään päde paremman ulkoasun suunnitteluun. Ulkoasun suunnittelu tieteellisen lähestymistavan jälkeen on suurimmaksi osaksi suunnittelijan inspiraatiota mutta myös testausta. Monien ideoiden toimivuutta ei pysty etukäteen näkemään, ja useimmiten suunnittelutyö onkin vain eri ideoiden pyörittelyä ja testaamista. Hyvän sivun suunnittelu ulkoasun näkökulmasta onkin onnistunut yhdistelmä tieteellistä testausta, inspiraatiota ja suunnittelijan tyylitajua sekä käyttäjien tarpeiden huomioimista. (Nielsen 2010, 12.)

4.2 Sivun suunnittelu

Ensimmäinen asia jonka sivun suunnittelija kohtaa on, mihin tarvittavat asiat sijoitetaan. Oma tilansa tarvitaan ainakin sisällölle ja navigaatiolle, mahdollisesti myös esimerkiksi mainoksille, haulle sekä toissijaiselle navigaatiolle. Nyrkkisäännön mukaan sisällölle täytyy varata vähintään 50 % alasta, mielummin jopa 80 %. Navigaation osuus ei tulisi olla yli 20 % ja se tulisikin pitää niin pienenä kuin se on järkevästi mahdollista. Käytettävyyden kannalta olisi parasta että sivulle ei sisällytettäisi ollenkaan lataamista raskauttavia mainoksia. Ne myös ohjaavat käyttäjän katsetta useimmiten väärään suuntaan. Tässäkin asiassa on hyvä muistaa

että käyttäjät tulevat sivulle tiedon houkuttelemana, joten mitä nopeammin se löydetään, sitä parempi. (Nielsen 2000, 19-22.)

Kun elementtien sijoittelua suunnitellaan, on hyvä muistaa että sen tulisi helpottaa nopeaa tiedon löytämistä ja omaksumista. Tiedon löytymisen nopeus on lähes täysin verrannollinen tiedon kokonaismäärään. Mitä yksinkertaisempi, sen parempi -periaate pätee erityisesti myös Internetissä. Jos käyttäjätestauksessa todetaan että ilman jotain elementtiä pärjätään, se on syytä poistaa kokonaan. Tiedon hakeminen monimutkaiselta sivulta on silmälle rasittavaa. Jokainen silmän liike tarvitsee silmän pysähdyksen ja siten pidentää tiedon löytymisen aikaa. Silmää voi myös ohjata löytämään olennainen tieto helpommin esimerkiksi käyttämällä taustavärejä tärkeissä elementeissä. Jos yhdellä vilkaisulla voidaan nähdä kaikki pääelementit ääreisnäön avulla, on sijoittelu tarpeeksi yksinkertainen. (Nielsen 2000, 22; Näsänen, 2007.)

Sivun sijoitteluun liittyy myös kysymys elementtien koosta. Sivua suunniteltaessa on mahdoton ennakoida, minkälaisella näyttölaitteella ja millä resoluutiolla sivua katsellaan. Myöskään näyttölaitteen kuvasuhde ei ole itsestäänselvyys, varsinkin kun mobiililaitteiden pystysuuntainen selaus kasvattaa osuuttaan. Tällä hetkellä mobiiliselausten osuus on alle 5 prosenttia kaikesta selauksesta, mutta on kasvanut ja tulee kasvamaan koko ajan. Suurin osa, eli noin 25 prosenttia selauksesta, tapahtuu 1024 kertaa 768 pikselin resoluutiolla. Vaikka suuremmat resoluutiot ovat työpöytäkäytössä kasvussa, myös toisaalta pienemmät resoluutiot ovat lisääntymässä mobiiliselausten myötä. (StatCounter 2011.)

Ideaalitapauksessa sivu suunnitellaan siten, että se ei ole riippuvainen mistään näyttölaitteen ominaisuudesta, vaan se pystyy mukautumaan tapaukseen kuin tapaukseen. Periaatteena tällaisessa suunnittelussa on, että mitään ei määritellä tiettyyn pikselikokoon, vaan koot määritellään prosentteina käytettävissä olevasta tilasta. Myös kirjasinkoot vaihtelevat käyttäjän asetuksista riippuen, ja se tulee ottaa huomioon jos suunnitellaan täysin skaalautuvaa sivua. Suunnittelijan visioille tällainen lähestymistapa aiheuttaa pieniä ongelmia, kun esimerkiksi valmiiksi piirrettyjä graafisia elementtejä täytyy välttää mahdollisimman pitkälle. Kannattaakin pidättäytyä pois pikseligrafiikasta mahdollisimman pitkälle. Vaihtoehtoina

ovat tyylimäärittelyt CSS:llä tai esimerkiksi skaalautuva vektorigrafiikka. (Nielsen 2000, 28-29.)

Joskus kuitenkin esimerkiksi sivun brändi edellyttää tiettyjä graafisia elementtejä ja silloin täytyy tyytyä esimerkiksi elementtien sijoitteluun käytettävissä olevan näyttöalan mukaan. Tämän lisäksi selkeiden graafisten symbolien lukeminen on käyttäjälle paljon nopeampaa kuin tekstimuotoisen informaation. Jos graafisia symboleita halutaan käyttää, täytyy kiinnittää huomiota niiden yksinkertaisuuteen sekä riittävään eriyvyyteen, jotta niiden tunnistaminen on nopeaa. Liikennemerkit edustavat klassista esimerkkiä aiheesta, jossa tieto olisi tekstimuodossa huomattavasti hitaampaa omaksua. (Nielsen 2000, 28-29; Näsänen 2007.)

Suurien graafisten elementtien pois jättämistä puoltaa kuitenkin toinenkin asia, joka on vasteaika. Sivun latausnopeus ja sitä kautta vasteaika on Nielsenin mukaan sivun suunnittelun tärkein osa-alue. Vasteajan tulisi olla alle sekunti, jotta käyttäjä voi tuntea liikkuvansa vapaasti. Sekunti on todella lyhyt aika, ja siihen pääseminen voi tuntua vaikealta, minkä vuoksi uusien nopeiden yhteyksien myötä tuntua houkuttelevalta unohtaa koko asia. Täytyy kuitenkin muistaa jälleen, että mobiiliselauksen lisääntyessä yhä useampi käyttäjä vierailee sivulla usein hyvin hitaiden langattomien yhteyksien varassa. Tämän takia latausnopeus pysyy edelleenkin relevanttina asiana. (Nielsen 2000, 42; StatCounter, 2011.)

4.3 Käyttöliittymäsuunnittelu

Käyttöliittymien suunnittelu verkkosivuja varten eroaa periaatteiltaan perinteisestä käyttöliittymäsuunnittelusta. Suurimpana erona suunnittelijan täytyy luopua täydestä kontrollista loppukäyttäjän suhteen. Perinteisemmissä käyttöliittymissä, niin fyysisissä sekä graafisissa, suunnittelija voi päättää, miten loppukäyttäjä hänen käyttöliittymäänsä käyttää. Verkossa kuitenkin suurin etu on vapaus, ja suunnittelijan täytyy jakaa vastuuta myös käyttäjälle. (Nielsen 1997.)

Perinteisemmässä suunnittelussa suunnittelija pystyy pakottamaan käyttäjän haluamilleen reiteille. Hän voi esimerkiksi sulkea reittejä piilottamalla painikkeita tai

pakottaa käyttäjän vastaamaan kysymykseen ennen eteenpäin pääsyä. Tämä ei kuitenkaan päde verkkosuunnittelussa. Verkossa käyttäjä päättää reittivalintansa, ja saattaa jopa luoda omia reittejä vaikka tulemalla suoraan tietylle sivulle hakukoneesta käymättä etusivun kautta. Jos suunnittelija yrittää tahallaan rajoittaa käyttäjää verkkosivulla tulee sivusta kankean oloinen, enemmän television tapainen passiivinen tiedonlähde kuin verkkosivun tapainen vapaa ja interaktiivinen lähde. Käyttöliittymäsuunnittelun tulee verkossa kunnioittaa vapautta ja edesauttaa sitä. (Nielsen 1997.)

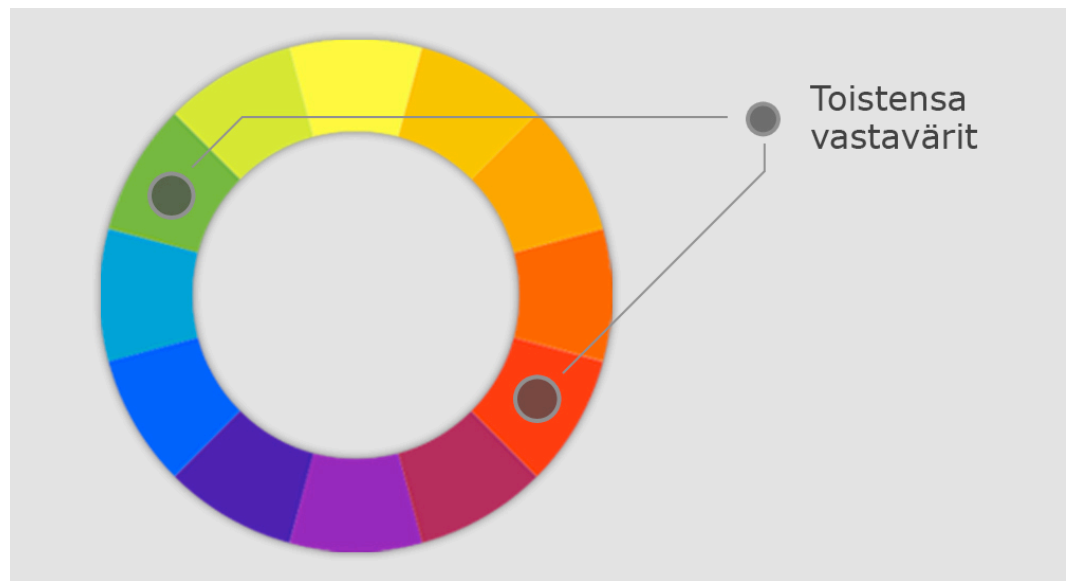
Kun käyttäjä opettelee uutta ohjelmistoa, hän käyttää verrattain paljon aikaa käyttöliittymän opetteluun ja oppii samalla kun käyttää sitä. Verkossa käyttäjä saattaa vieraillla sivulla vain vaikka minuutin, joten uuden opettelu on haasteellista. Tämän takia verkkosuunnittelussa korostuu jälleen äärimmäinen yksinkertaisuus sekä hyväksi havaittujen käytäntöjen käyttäminen. Saapuessa sivulle käyttäjän on jo osattava käyttää sitä, tai hänen mielenkiintonsa sivun tarkasteluun saattaa loppua. On myös muistettava, että yksittäisellä ohjelmalla voi olla oma hyväksi havaittu logiikkansa ja tapa tehdä eri asioita. Verkossa kuitenkin korostuu ajatus että koko WWW on yksi ohjelma. Tämän takia verkkosivulla ei ole syytä yrittää tahallaan eriytyä muista sivuista käyttöliittymän osalta. Tietenkin ilman uuden kokeilua verkon kehitys pysähtyy, mutta on syytä miettiä tarkkaan sivuja suunniteltaessa, kuinka paljon uutta käyttäjä on valmis opettelemaan. (Nielsen 2000, 271.)

4.4 Värit

Värien käyttö ja teoria WWW-ympäristössä nojautuu vahvasti samoihin sääntöihin kuin perinteisemmissä medioissa. Koska WWW-sivustot ovat lähtökohtaisesti tehty esittämään tietoa käyttäjälle tekstin muodossa, on riittävä kontrasti tärkeimpiä värisuunnittelun osia. Suurin kontrasti saavutetaan luonnollisesti mustalla tekstillä valkoisella pohjalla, tai toisin päin. Kumpi näistä on parempi vaihtoehto, on pitkälti makukysymys. On esitetty että musta teksti valkoisella pohjalla on raskavampaa silmille, koska tietokoneen näytöissä valkoinen on kirkkain osuus. Toisaalta jotkut kokevat että valkoisen tekstin lukeminen mustalta pohjalta rasittaa enemmän silmiä ja esimerkiksi näyttölaitteen heijastukset näkyvät selvemmin

tummalla taustalla. Yleensä ottaen molemmat ovat hyviä vaihtoehtoja ja valinnan määrää enemmän sivuston muu ilme ja tavoiteltu vaikutelma. (Näsänen 2007.)

Pelkän mustavalkoisuuden lisäksi kontrastia voi etsiä esimerkiksi vastaväreistä. Vastaväri tarkoittaa väriä, joka on suoraan valitun värin vastakkaisella puolella väriympyrällä. Näin ollen esimerkiksi vihreän vastaväri olisi oranssi. Sen lisäksi, että vastavärit sointuvat keskenään hyvin, niillä on vahva kontrastisuhde. (Wilder 2000.)



Kuvio 3 Väriympyrä ja vastavärit (Lindén 2011.)

Kontrastin tarvetta voi vähentää suurentamalla näkyvää elementtiä, kuten esimerkiksi tekstiä. Pienet yksityiskohdat piirtyvät silmään heikkokontrastisempana kuin isot, mikä johtuu valon leviämisestä silmällä ja se aiheuttaa epäterävyyttä erityisesti reunoilla. Tästä johtuu kontrastin ja koon välinen suhde. Silmän optiikka yrittää korjata tätä epäterävyyttä itse, mutta se pätee ainoastaan vaaleuserojen havaitsemiseen. Emme siis havaitse kovinkaan hyvin pieniä elementtejä, kuten tekstiä, jos se erotetaan taustastaan ainoastaan värisävyllä ja vaaleusaste on sama. Tämä on hyvä muistaa värien valinnassa. Ääriesimerkkinä on vaaleanpunainen teksti vihreällä pohjalla: punavihersokeat eivät erota tekstiä ollenkaan. (Näsänen, 2007; Nielsen 2000, 126.)

4.4.1 Värien valinta

Kun sivua ja sen käyttöliittymää suunnitellaan, kannattaa se tehdä mustavalkoisena. Näin varmistetaan, että vaikka käyttäjä ei erottaisi värisävyjä ollenkaan, on sivu käytettävä. Väriä ei saisikaan käyttää ainoana symbolina millekään elementille, vaan sen kanssa tulisi käyttää myös jotain muuta dimensiota kuten muotoa tai kokoa. (Asplund 2010.)

Värien kokonaismäärä kannattaa yleensä pitää suhteellisen pienenä varsinkin, jos värejä käytetään korostamaan merkitystä ja käyttäjältä odotetaan sen muistamista. Tämän lisäksi usein osa väreistä määrittyy jo etukäteen esimerkiksi brändin kautta tai muuten asiakkaan pyynnöstä. Se edelleen pienentää valittavien värien määrää. Tämän jälkeen on syytä tarkkaan pohtia, mitä värejä käyttää ja mihin. Liian monta varsinkin kirkasta väriä sekoittaa silmää, eikä käyttäjä tiedä, mihin kohdistaa katseensa. (Hunt 2006.)

Kirkkaat värit ovat myös oikein käytettynä erittäin tehokkaita. Niitä on hyvä käyttää tehokkaana huomionherättäjänä. Perinteisesti niitä käytetään varoitusväreinä tai muistutusväreinä, mutta ne ovat hyviä esimerkiksi otsikoihin, jos tarkoitus on kiinnittää lukijan huomio nopeasti. Kirkkaisiin väreihin sisältyy kuitenkin se vaara, että väärin käytettynä ne vain ohjaavat katsetta vääriin suuntiin ja tämä turhauttaa käyttäjää. Edes yhden kirkkaan värin käyttö on kuitenkin suositeltavaa, koska ilman sitä sivun yleisilmeestä helposti puuttuu tarvittava värisyvyys ja kontrasti. (Hunt 2006.)

Kirkkaista väreistä tehokkaimmin toimii punainen ja heikoiten sininen, sillä ihmisen silmässä on vähiten sinistä aallonpituutta havaitsevia reseptoreja. Tämän lisäksi kirkas sininen on perinteisesti varattu linkkien väriksi. Sinisen merkitys linkkivärinä on vähenemään päin, mutta käyttäjät osaavat edelleen yhdistää sen automaattisesti. (Hunt 2006; Nielsen 2000, 64.)

4.5 WWW-Typografia

Typografia eli kirjasintaide on vanha taiteenlaji, jossa eri ominaisuuksien avulla pyritään saavuttamaan tekstimuotoisesta informaatiosta mahdollisimman helppolukuista ja samalla visuaalisesti miellyttävä kokonaisuus. Näitä eri ominaisuuksia ovat esimerkiksi kirjasintyyli, pistekoot, värit ja välistykset. WWW-typografia pohjautuu edelleen näihin muinaisiin sääntöihin, mutta tuo samalla sekä rajoituksia että mahdollisuuksia suunnittelijoiden saataville. (Laak 2006.)

Typografian merkitys perinteisissä julkaisuissa on ymmärretty jo pitkään. Sen sijaan WWW-typografiaan on kunnolla alettu kiinnittää huomiota vasta viime vuosina. Typografia on olemassa sen takia, että sisältö saadaan katsojalle mahdollisimman tehokkaasti ja miellyttävästi ja kuten Jakob Nielsen muistuttaa, tulevat käyttäjät sivustoille vain sisällön houkuttelemina. Kaikki muu ympärillä on taustaa, mikä mahdollisesti auttaa tai vaikeuttaa sisällön hahmottamista. Lieneekin selvää, että typografia on tärkein yksittäinen tekijä sisällön hahmottamisessa. (Nielsen 2000, 99.)

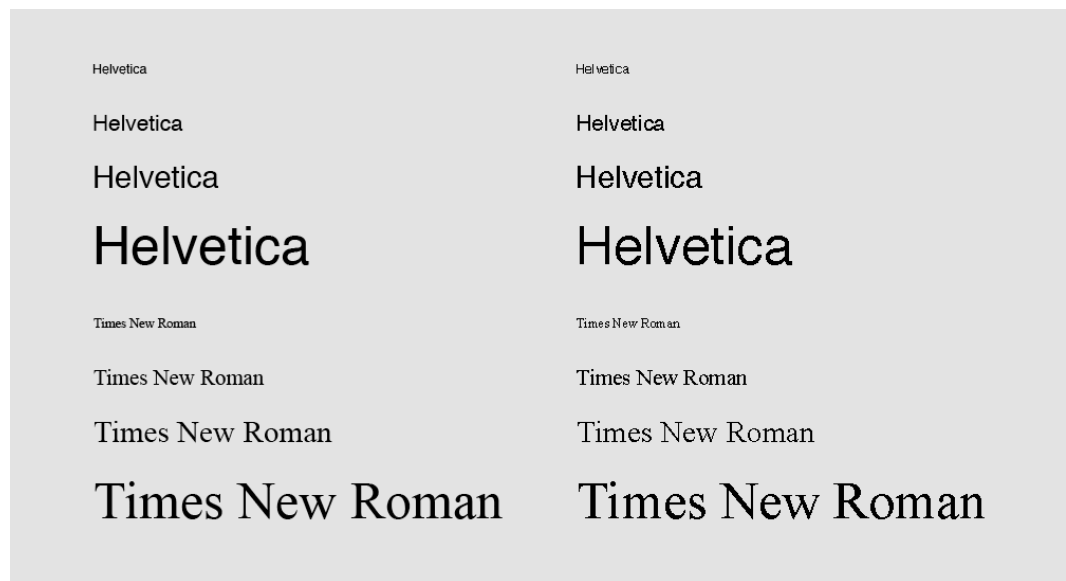
4.5.1 Typografian saavutettavuus

Perinteisessä typografiassa kirjasinten painamisen jälkeen niitä ei voi enää muuttaa. Se antaa suunnittelijalle täyden vallan sen suhteen, miltä lopullinen informaatio näyttää. Suunnittelija pystyy ennalta määäämään niin kirjasinten koon, värin kuin minkä tahansa ominaisuuden. Vastuu siirtyy sen jälkeen suoraan lukijalle, jos suunnittelija on osannut asiansa, ongelmaa ei ole. (Laak, 2006.)

WWW-typografiassa asia ei ole näin yksinkertainen. Suunnittelijalla on mahdollisuus tietenkin vaikuttaa etukäteen asioihin paljonkin. Hän voi määritellä kirjasimia sekä kirjasinten kokoja, mutta koska laitteet, joilla sitä katsellaan vaihtelevat rajusti, on otettava huomioon erityisiä piirteitä. (Laak 2006.)

Lukijoiden käyttämien näyttölaitteiden ominaisuudet vaihtelevat rajusti. Esimerkiksi tarkkuus perinteisessä painojäljessä voi olla jopa 2400 dpi, kun taas digitaal-

lisessa ympäristössä joudutaan tyytymään laajalti alle 300 dpi:n tarkkuuteen. Suurin osa näyttölaitteista toimii vain 96 dpi:n tarkkuudella. Tämä tekee rajoituksia tekstin kokoon ja sitä kautta luettavuuteen. Toinen digitaalisen tekstin luettavuuteen suuresti vaikuttava asia on reunojen pehmenneys. Vaikka reunojen pehmenneys onkin jo isossa osassa käyttöjärjestelmiä oletuksena päällä, se ei kuitenkaan ole esimerkiksi Windows XP:ssä, jota edelleen käytetään laajalti. Sen lisäksi että reunojen pehmenneys vaikuttaa positiivisesti kirjasinten ulkonäköön, ja se myös parantaa pienten kokojen luettavuutta runsaasti. (Laak 2006.)



Kuvio 4 Esimerkki reunojen pehmennyksestä (Lindén 2011.)

WWW-typografian muuttuvuus tarjoaa myös sen parhaat puolet. Näkörajoitteinen ihminen joutuu turvautumaan perinteisissä julkaisuissa erillisiin apuvälineisiin kuten suurennuslasiin ja aina sekään ei auta, jos painettu teksti on erityisen pientä, tai jos lukija on sokea. Internetissä teksti on kuitenkin valmiiksi jo digitaalisessa muodossa ja sitä on helppo muokata, kopioida sekä siirtää toisiin laitteisiin. Sen kokoa voidaan muuttaa, tai se voidaan esimerkiksi siirtää kokonaan näkörajoitteisten lukijalaitteelle. Näin ollen digitaalinen teksti ei ole läheskään niin rajoittunutta kuin perinteinen painettu typografia, ja se on selkeästi saavutettavampaa. (Luuk 2006.)

Tekstin luettavuuteen vaikuttaa myös sen asemointi sivulla. Länsimaissa on totuttu lukemaan tekstiä vasemmalle tasattuna, ja jos ei ole todella hyvää syytä, kannattaa se niin pitää. Teksti saa ilmettä, jos se tasataan poikkeavasti joko keskelle tai oikealle, mutta tällä tavoin sen lukunopeus laskee selkeästi. (Nielsen 2000, 126.)

4.5.2 Fontin valinta

Kirjasintyyppin valinta on ensimmäinen askel sivun typografian suunnittelua. Se pitkälti määrittelee sivun yleisilmeen ja muut valinnat. Perinteisesti WWW-typografia kärsii valinnan mahdollisuuksien puutteesta, kun yleisesti käytettyjä ja turvallisia kirjasintyyppejä ei ole montaa. Siksi aivan ensiksi onkin tärkeää tunnistaa olemassa olevat vaihtoehdot. (Hume 2005.)

WWW-typografiassa on käytännössä kaksi eri kirjasintyyppien ryhmää: serif sekä sans-serif. Serif tarkoittaa kirjasinten pieniä lisäviivoja sekä ylimääräisiä koriste-
luja päälinjojen päätepisteissä. Perinteisesti ajatellaan, että nämä pienet lisäviivat auttavat lukijan silmää siirtymään kirjaimesta toiseen, ja serif-kirjasintyypit onkin erittäin suosittuja painetussa tekstissä. Sans-serif taas yksinkertaisesti tarkoittaa kirjasintyyppejä ilman näitä lisäviivoja. Ne tunnetaan myös nimellä groteski. (Hume 2005.)

Serif-fontit aiheuttavat kuitenkin pientä ongelmaa WWW-suunnittelussa varsinkin pienikokoisina, koska pienet yksityiskohdat tekevät kirjasimesta epäselvän. Tämän lisäksi reunojen pehmennys saattaa aiheuttaa ongelmia. Joskus sivun sisältö vaatii serif-tyylisen kirjasimen esimerkiksi kun haetaan vanhaa tai perinteistä vaikutelmaa. On kuitenkin hyvä ymmärtää, miksi monet suunnittelijat päätyvät usein groteskiin vaihtoehtoon. (Hume 2005.)

Verdana

Sans-serif kirjasin

Georgia

Serif kirjasin

Kuvio 5 Esimerkit Sans-serif ja Serif kirjaksimista (Lindén 2011.)

Hyvän kirjasiimen WWW-suunnittelua varten tunnistaa muutamasta eri seikasta: yleisilmeeltään selkeä ja yksinkertainen, tarpeeksi tilaa kirjainten välissä, tarpeeksi tilaa kirjainten sisällä sekä pitkä x-korkeus. X-korkeudella tarkoitetaan tässä matkaa kirjasiimen peruslinjasta keskilinjaan. Nämä vaatimukset täyttävät mm. paljon käytetyt Verdana, Trebuchet MS sekä Helvetica. Tietysti kaikissa kirjasiimissa eivät kaikki ominaisuudet täydellisesti kohtaa, mutta ne ovat hyvä lähtökohta. (Hume 2005.)



Kuvio 6 Hyvän WWW-kirjasiimen piirteitä (Lindén 2011.)

5 CASE

5.1 Lähtökohdat ja tavoitteet

Case-projektina toimi tässä opinnäytteessä Tonttipörssi-palvelun konsepti. Tavoitteena oli suunnitella ja rakentaa toimiva konsepti uudeltaisesta palvelusta jolla voitaisiin markkinoida pienten kuntien uusia asuntoalueita sekä myydä tontteja. Projektityöryhmässä toimi itseni lisäksi Ville Pirskanen Ohjelmistotekniikasta sekä Essi Carlson Miljöosuunnittelusta. Pilottialueeksi palvelun sisältöä varten valittiin Kärkölen kunnan Lappilan kylässä sijaitseva Louhivaara.

Kun Tonttipörssi-palvelua alettiin suunnitella, asetettiin sille muutamia perusvaatimuksia: saavutettavuus, helppokäyttöisyys, uutuusarvo sekä monistettavuus. Nämä linjaukset määrittivät pitkälti suunnitteluprosessia.

Eri toteutustekniikoita arvioitiin reilusti ja ensimmäinen suuri päätös oli jättää Flash kokonaan pois palvelusta. Vaikka palvelu tulisi sisältämään erilaisia rikkaan sisällön elementtejä ja haluttiin hakea uudenlaista käyttöliittymää, todettiin Flash kuitenkin hieman saavuttamattomaksi sekä eilispäivän tekniikaksi. Täytyy huomioida, että nimenomaan tämän palvelun toteutuksessa se oli mahdollista, koska pystyimme toteuttamaan kaiken tarvitsemamme ilman sitä. Monessa muussa tilanteessa valinta olisi varmasti ollut erilainen. Tässä tapauksessa pärjättiin hyvin JavaScriptillä, php:llä sekä Googlen tarjoamilla rajapinnoilla.

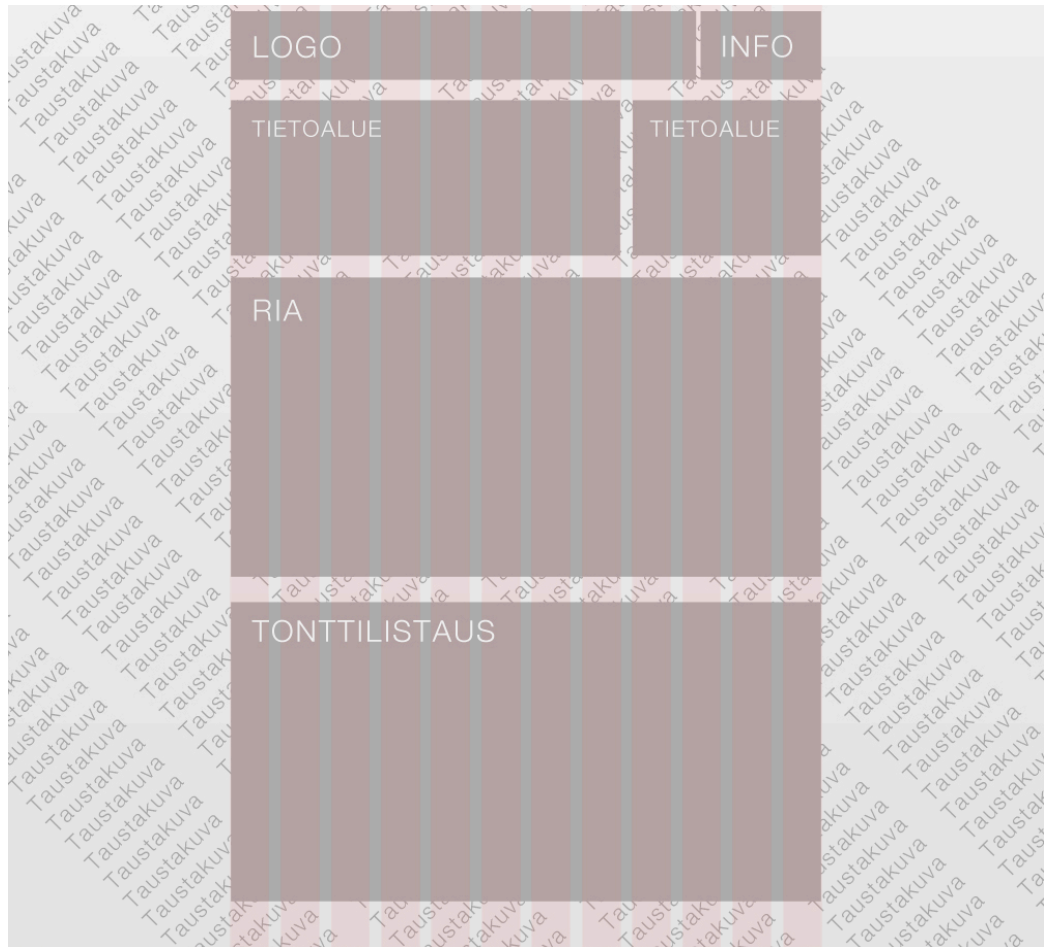
Teknisessä toteutuksessa suurimmiksi haasteiksi nousi Googlen tarjoamien karttapalveluiden yhdistäminen saumattomaksi osaksi palvelun käyttöliittymää sekä palvelun mahdollisimman helppo päivitettävyyden. Koska projektissamme oli mukana myös ohjelmoinnin asiantuntija, päädyimme ratkaisuun jättää valmiit julkaisujärjestelmät pois prosessista. Asiantuntijana toiminut Pirskanen ohjelmoi itse palveluun hallintapaneelin, josta asiakkaan edustaja pystyisi helposti päivittämään tarvittavat tiedot.

Tämän opinnäytteen vastuualueella oli palvelun käyttöliittymän suunnittelu yhdessä Pirskasen kanssa, visuaalinen suunnittelu sekä front-end-ohjelmoinnin toteuttaminen. Suunnitteluun otettiin tärkeimmäksi lähtökohdaksi dynaamisuus. Sisältö sekä ulkonäölliset seikat ovat sivun päivittäjän muutettavissa, joten se ei saa vaikuttaa negatiivisesti yleisilmeeseen. Myös joidenkin sisältöjen kokoa ei voi etukäteen tietää, joten sekin olisi otettava huomioon suunnittelussa.

5.2 Työn kuvaus

5.2.1 Verkkopalvelun suunnittelu

Palvelun rakenne on yksisivuinen, ja kaikki sisältö esitetään joko välilehdillä tai erillisillä ikkunoilla. Tähän ratkaisuun päädyttiin, koska sisällön jakamiseen eri sivuille ei nähty tarvetta, vaan kaikki sisältö olisi nähtävissä kerralla. Tähän ei kuitenkaan täysin päästy pienillä näyttöresoluutioilla mutta kompromisseja oli tehtävä. Alueen kartta haluttiin nostaa pääosaan sivulla ja samasta kohtaa löytyy syventäviä tietoja alueesta välilehtien takaa. Näin sisällöt saatiin järjesteltyä omiin osioihin luonnollisesti, ilman erillisiä alisivuja. Välilehtimäinen rakenne on myös siksi hyvä valinta sisällölle, koska siitä on helppo jättää osioita pois kun palvelua räätälöidään eri tarpeisiin. Eri osioiden korkeus yritettiin pitää mahdollisimman pieninä, että sisältö mahtuisi käyttäjän näkymään kerralla. Leveys puolestaan määräytyi valitusta 960s-ruudukkojärjestelmästä. Siinä sivu jaetaan leveyssuunnassa yhtä suuriin palkkeihin ja näin saavutetaan järjestelmällisyyttä ja helppoutta eri elementtien leveyksissä. Valitussa järjestelmässä ruudukko koostuu 12:sta 60 pikseliä leveästä palkista. Palkkien ympärillä on sen lisäksi 10 pikselin marginaalit, ja näin koko sivun leveydeksi määräytyy 960 pikseliä, joka toimii hyvin suurimmalla osalla käytössä olevista näyttöresoluutioista.

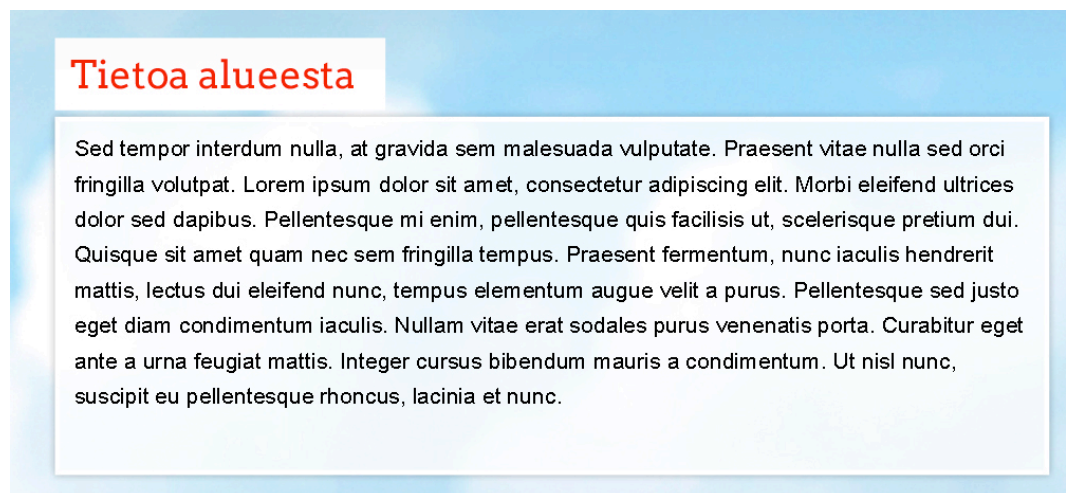


Kuvio 7 Palvelun rautalankamalli (Lindén 2011.)

Visuaalisessa suunnittelussa tavoiteltiin neutraalia yleisilmettä, jota olisi helppo räätälöidä kunkin kunnan tarpeisiin. Tämä saavutettiin käyttämällä paljon neutraalia valkoista sekä harmaansävyjä. Varsinaisen ulkonäön sivulle tuo yksi määriteltävä tehoväri, joka oli Kärkölen tapauksessa puhdas punainen, sekä iso vaihdettava taustakuva. Tasaiset väripinnat tuovat tarvittavaa rauhallisuutta sisällölle ja tasapainottaa taustakuvan tuomaa visuaalista ärsykettä. Tasaiset väripinnat olivat myös selvä valinta sen takia, että erillisiä ladattavia kuvia voitaisiin käyttää mahdollisimman vähän ja väripinnat voitaisiin määritellä tyyli tiedostossa.

Saavutettavuutta pyrittiin luomaan teknisten valintojen lisäksi visuaalisessa suunnittelussa. Kontrastit värien, pintojen ja sisällön välillä pyrittiin pitämään korkeana, ja tekstisisällössä päädyttiinkin turvalliseen mustaan tekstiin valkoisella pohjalla. Kontrastin lisäksi tärkeimmäksi asiaksi saavutettavuuden saralla visuaali-

suuden näkökulmasta nousi typografia. Palvelun typografiassa pyrittiin noudattamaan hyvän internet-typografian periaatteita. Leipätekstin kirjasimeksi valittiin groteski kirjasintyyppi. Tämän lisäksi otsikoissa ei haluttu käyttää mitään liitännäispohjaisia ratkaisuja, jotta ne olisivat mahdollisimman luettavia. Tässä kohtaa päädyttiinkin Googlen tarjoamaan Web Fonts API:in ja sen fonttien käyttöön. Web Fonts API:n avulla voidaan laajentaa käytettävissä olevien fonttien määrää helposti, ja käyttämättä mitään erillisiä liitännäisiä. Tarjolla olevien fonttien avulla saadaan palvelulle erottuvampaa ilmettä.



Kuvio 8 Esimerkki tekstisisällön esittämisestä (Lindén 2011.)

Palvelun helppokäyttöisyys oli yksi tärkeimpiä lähtökohtia suunnitteluprosessille. Työryhmässä haluttiin kuitenkin myös korostaa uutuusarvoa, ja näiden kahden yhdistämisessä syntyy yleensä pieniä konflikteja. Helppokäyttöisyys tarkoittaa useimmille opittujen käytäntöjen käyttöä, joten kaikki uuden opettaminen käyttöliittymässä on pieni riski. Palvelun käyttöliittymää haluttiin kuitenkin tuoda lähemmäs työpöytäsovellusta, ja perinteisestä WWW-sivuston rakenteesta pyrittiin tietoisesti pois.

Helppokäyttöisyyttä pyrittiin parantamaan käyttämällä internetin käyttäjille tuttuja elementtejä, kuten Googlen suosittua karttapalvelua. Vaikka tavoitteena olikin uudenlainen tapa saada tietoa tonteista, yritettiin käyttöliittymän eri osiot säilyttää mahdollisimman tutun oloisina. Elementtien ulkonäköä muokattiin jonkin verran,

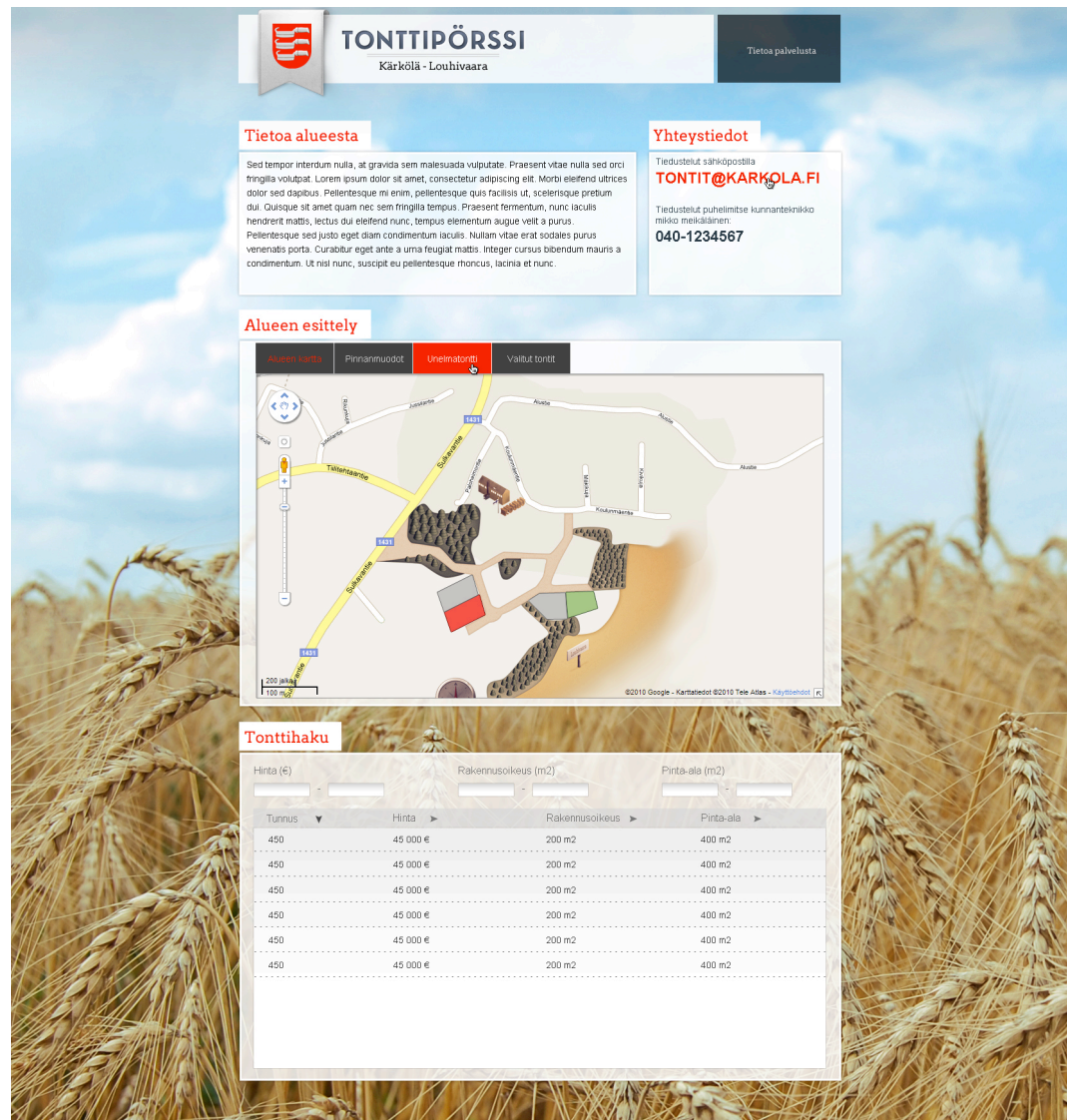
jotta saataisiin palvelulle oma ilme, mutta samalla pidettiin mielessä niiden tunnistettavuus, jotta käyttäjän ei tarvitsisi käyttää aikaa uuden opetteluun.

Uutuusarvo on ominaisuus, jota jollain tasolla tavoitellaan lähes aina uusien palveluita suunniteltaessa. Tässä projektissa oltiin kuitenkin valmiita jopa uhraamaan muita alueita uutuusarvon takia. Koko projektin tarkoitus oli löytää uusia ja raikaita ideoita, miten tontteja voisi ylipäästänsä markkinoida verkkoympäristössä. Siksi myös tämä ominaisuus nousi erittäin tärkeäksi.

Lopuksi palvelulle suunniteltiin myös tunnus. Tunnukseen haluttiin sisällyttää helppolukuisuutta sekä ajattomuutta. Näiden seikkojen varmistamiseksi valittiin tunnuksen pohjaksi groteski kirjasin, ja sitä muotoiltiin uusiksi Illustrator ohjelmalla. Kun haluttu muoto oli saavutettu, lisättiin siihen hillityt liukuvärit ja syvyyssefektit, jotta se näyttäisi mielenkiintoisemmalta verkkopalvelussa.

The logo consists of the word "TONTTIPÖRSSI" in a bold, blue, sans-serif typeface. The letters are slightly irregular, giving it a hand-drawn or 'grotesque' feel. Below the text, there are two thin, horizontal, slightly wavy lines that act as a base or underline for the word.

Kuvio 9 Palvelun tunnus (Lindén 2011.)

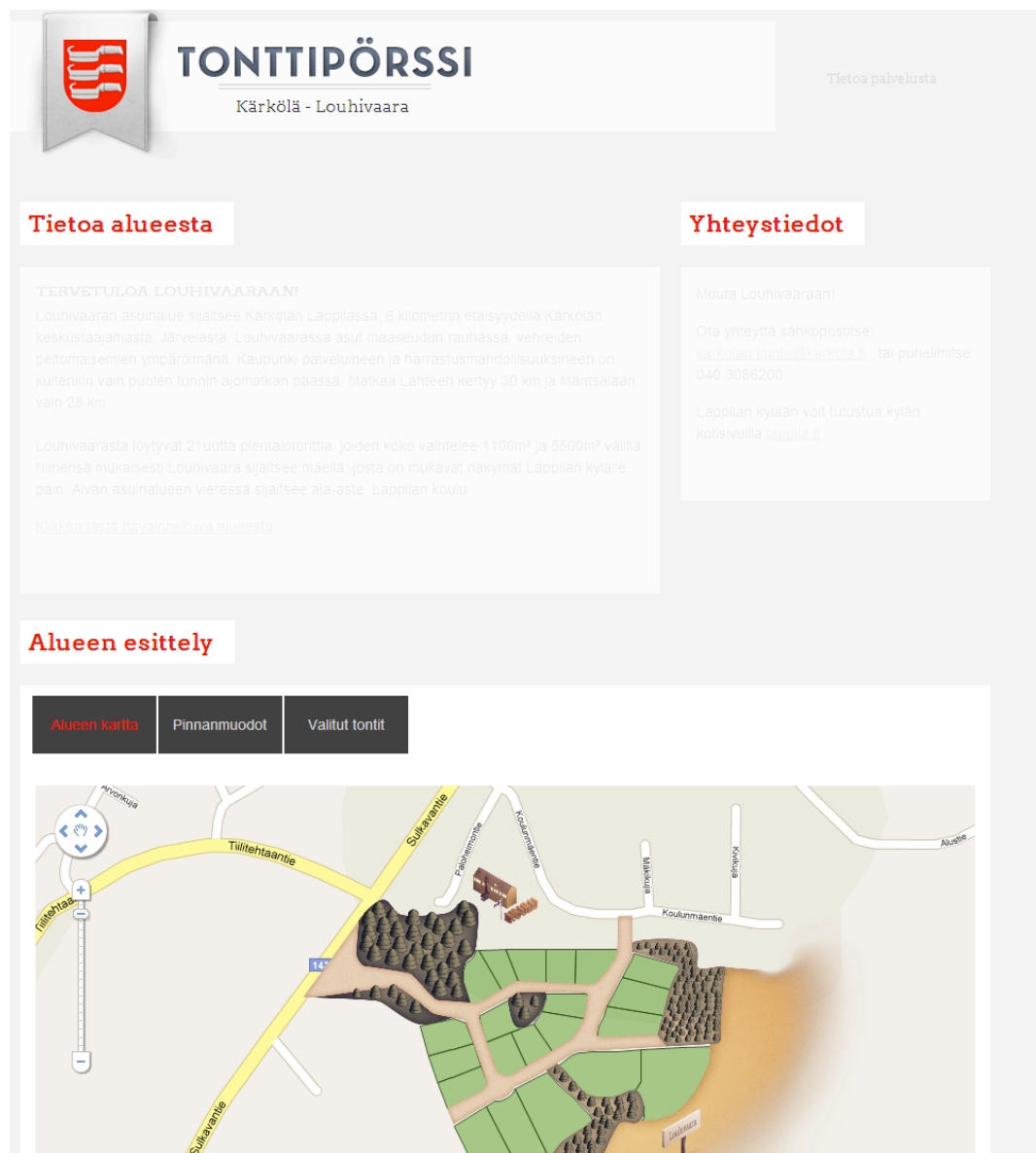


Kuvio 10 Valmis ulkoasusuunnitelma (Lindén 2011.)

5.2.2 Verkkopalvelun tekniikka

Tämän opinnäytteen alueelle kuului visuaalisen suunnittelun lisäksi palvelun toteutus front-end kehityksen näkökulmasta. Palvelun toteutukseen tarvittiin HTML, CSS ja JavaScript osaamista. Koska konseptin oli tarkoitus myös olla uudenaikainen, päädyttiin käyttämään CSS3:a niin paljon hyödyksi kuin mahdollista. Kuten aikaisemmin jo tutustuttiin, CSS3:a tuetaan jo laajasti mutta erityisesti Microsoftin Internet Explorer aiheutti ongelmia.

Selainten erot olivat suuressa osassa toteutusprosessia. Suurimmalla osalla nykyaikaisia selaimia ei ole mitään ongelmia ymmärtää normaaleja CSS-käytäntöjä. Niillä elementtien ulkonäkö on juuri sellainen, kuin on suunniteltu. Suurimmaksi ongelmaksi muodostui Internet Explorer, ja toteutuksessa tehtiin päätös olla tukematta Internet Explorerin versionumero seitsemää vanhempia selaimia. Sen kanssa tuli ongelmia esimerkiksi elementtien sijoittelun kanssa, elementtien läpinäkyvyyksien kanssa ja ylipäätään CSS3:n kanssa. Ongelmien korjaamiseksi päädyttiin kirjoittamaan Explorer-selaimille oma tyylimäärittely.



Kuvio 11 Ongelmia Internet Explorerilla ennen korjauksia (Lindén 2011.)

JavaScript osuuksissa hyödynnettiin eri jQuery-kirjastoja. JQuery:n käyttöönotto on helppoa, ja siihen löytyy valmiina lisäosat lähes tarkoitukseen kuin tarkoitukseen. Lisäksi sen käyttö vaatii todella vähän tai ei ollenkaan varsinaisen JavaScript syntaksin tuntemista.

Toteutuksessa käytettiin jQuery TOOLS-käyttöliittymäkirjastoa. Siihen on kerätty yleisimmät nykyaikaisen verkkosivun tarvitsemat jQuery:n osat. Sen avulla voidaan esimerkiksi muodostaa välilehtirakenteita, helppokäyttöisiä lomakkeita, tietokuplia sekä kerrostuvia ikkunoita. JQuery TOOLS:n käyttöönotto on vieläkin yksinkertaisempaa kuin alkuperäisen JQuery:n, koska siinä voi valita kaikki tarvitsemansa osat yhteen tiedostoon. Sen toimivuus on myös taattu kaikilla selaimilla, jopa Internet Explorerin kuudennesta versiosta eteenpäin, joten sen käyttö on hyvinkin perusteltua. Tässä tapauksessa sitä käytettiin lisätietoikkunan tekemiseen Overlay-lisäosalla, välilehtirakenteen muodostamiseen esittely-osioon Tabs-lisäosalla sekä kolmiulotteisen kartan esittelyn tekemiseen Slideshow-lisäosalla. Näiden käyttöönotossa ei ilmennyt mitään ongelmia, ja ne toimivat suhteellisen hyvin kaikissa testatuissa selaimissa. Taustakuvan skaalaamiseen kokeiltiin myös erilaisia jQuery-lisäosia. Niiden kanssa oli hieman ongelmia, koska ne osoittautuivat hyvin raskaaksi suorittimelle.

Välilehtirakenteiden tuottaminen sujui yksinkertaisesti. JQuery TOOLS:n verkkosivulta ladattiin tarvittava paketti, johon sisällytettiin JQuery:n ydin, Tabs-lisäosa sekä Tabs:n laajennus Slideshow. Kaikki eri osat sisältyvät yhteen JavaScript-tiedostoon, ja se linkitetään verkkosivulle samaan tapaan kuin muutkin JavaScript-tiedostot.

HTML:ssä täytyy määritellä luokkien avulla kehys, jonka sisään muodostetaan div-tagien avulla välilehdet. Jokainen div-tag edustaa yhtä välilehteä. Tämän lisäksi täytyy luokkien avulla määritellä myös seuraava- sekä edellinen-painike, ja suorat linkit eri välilehdille. Kun perusrakenne on saatu muodostettua, on sen muokkaaminen omiin tarpeisiin joustavaa ja helppoa.

Tämän palvelun tapauksessa välilehtirakenteita upotettiin sisäkkäin useita. Näin saatiin muodostettua välilehtirakenne, jonka eri välilehdillä on vielä omat raken-

teensa. Tätä ideaa hyödynnettiin esimerkiksi 3D-kartan slideshow-esityksessä. Slideshowssa noudatettiin yleisesti hyväksyttyä tyyliä, jossa painikkeet ovat dioiden molemmilla puolilla, ja pikalinkit eri sivuille sijaitsevat alapuolella pieninä palloina.

YHTEENVETO

Nykyaikaisen verkkopalvelun suunnittelu ja toteutus on monimutkainen prosessi. Palveluiden vaatimustaso nousee jatkuvasti, ja niiden olisi tarkoitus haastaa työpöytäsovellukset ja jopa nousta niiden ohi. Kuitenkin verkkoon suunniteltaessa on muistettava sääntöjä jotka pakottavat yksinkertaisuuteen sekä mutkattomuuteen. Näiden kahden asian yhdistäminen voi olla todella haastavaa. Miten suunnitella palvelu, jonka käytön yksinkertaisuudessaan oppii välittömästi, mutta samalla tarjoaa työpöytäsovelluksen monipuolisuuden. Haasteita riittää myös teknisten alustojen kirjavuudessa. Tällä hetkellä suunnittelija ei voi mustavalkoisesti luottaa että palvelua käytetään juuri tietynlaisella laitteella ja tietynlaisessa tilanteessa. Siksi suunnitteluprosessi täytyy pitää mahdollisimman dynaamisena, ja mitään ei kannata pitää liian ehdottomana.

Kaikista tärkeimmäksi osaksi tämänkaltaista palvelua tehtäessä muodostuu suunnittelu. Suunnittelun tärkeyttä ei voi tarpeeksi alleviivata. Jos koko palvelu, sen tekniset yksityiskohdat ja sisältö, on hyvin suunniteltu, on varsinainen toteuttaminen vain suoritus. On myös erittäin hyödyllistä tutustua erikseen tarjolla oleviin valmiisiin vaihtoehtoihin, mitä esimerkiksi Google tarjoaa. Oikein käytettynä ne voi säästää tunteja ylimääräistä työtä sekä tarjota loppukäyttäjälle syvemmän ja mukavamman käyttökokemuksen.

Käyttökokemuksen mutkattomuuteen on syytä keskittyä tarpeeksi suunniteltaessa verkkopalvelua. Kaikki työryhmän suunnittelemaat uudenaikaiset ominaisuudet menevät hukkaan, jos loppukäyttäjä ei osaa käyttää niitä. Tämän lisäksi oppimisen pitäisi tapahtua todella nopeasti. Täytyykin miettiä tarkoin tasapainoa ominaisuuksien monipuolisuuden ja yksinkertaisuuden välillä. Myös kokonaisuudesta täytyy saada yksinkertainen sekä johdonmukainen, ettei se ole vain epämääräinen kasa erilaisia ominaisuuksia. Palvelun täytyy olla selkeä ja yksinkertainen kokonaisuus.

LÄHTEET

1. Painetut lähteet

Cederhol, D. 2010. CSS3 For Web Designers. New York: A Book Apart.

Chaffer, J. & Swedberg, K. 2009. Learning jQuery 1.3. Birmingham: Packt Publishing Ltd.

Jakob, N. 2000. WWW suunnittelu. IT Press.

Keith, J. 2010. HTML5 For Web Designers. New York: A Book Apart.

2. Sähköiset lähteet

Allaire, J. 2010. [Verkkolähde] The Future of Web Content – HTML5, Flash & Mobile Apps [viitattu 15.1.2011]

Saatavissa: <http://techcrunch.com/2010/02/05/the-future-of-web-content-html5-flash-mobile-apps/>

Asplund, M. 2010. [Verkkolähde] Värien säännöt [viitattu 20.02.2011]

Saatavissa: reppu.lamk.fi

Google. 2011. [Verkkolähde] Google Maps JavaScript API V3 [viitattu 10.01.2011]

Saatavissa: <http://code.google.com/apis/maps/documentation/javascript/>

Hume, A. 2005. [Verkkolähde] The Anatomy of Web Fonts [viitattu 14.1.2011]

Saatavissa: <http://articles.sitepoint.com/article/anatomy-web-fonts>

Hunt, B. 2006. [Verkkolähde] Colour in Web Design [viitattu 25.2.2011]

Saatavissa: <http://www.webdesignfromscratch.com/web-design/colour/>

Jobs, S. 2010. [Verkkolähde] Thoughts on Flash [viitattu 15.1.2011]

Saatavissa: <http://www.apple.com/hotnews/thoughts-on-flash/>

Laak, T. 2006. [Verkkolähde] Saavutettavaa typografiaa – Osa 1 [viitattu 13.1.2011]

Saatavissa: <http://saavutettava.fi/2006/03/24/saavutettavaa-typografiaa-osa-1/>

Mary. 2010. [Verkkolähde] Silverlight vs. Flash: How Silverlight Compares to Flash [viitattu 18.1.2011]

Saatavissa: <http://blog.sherweb.com/silverlight-vs-flash-how-silverlight-compares-to-flash/>

Nielsen, J. 1997. [Verkkolähde] The Difference Between Web Design and GUI Design [viitattu 10.3.2011]

Saatavissa: <http://www.useit.com/alertbox/9705a.html>

Nielsen, J. 2000. [Verkkolähde] Flash: 99% bad [viitattu 19.12.2010]

Saatavissa: <http://www.useit.com/alertbox/20001029.html>

Näsänen, R. 2007. [Verkkolähde] Visuaalisen käytettävyyden opas 2007 [viitattu 10.2.2011]

Saatavissa:

http://www.ttl.fi/fi/ergonomia/kognitiivinen_ergonomia/visuaalinen_kaytettavyys/Documents/Visuaalisen_kaytettavyiden_opas_2007.pdf

StatCounter. 2011. [Verkkolähde] StatCounter Global Stats [viitattu 10.2.2011]

Saatavissa: <http://gs.statcounter.com/#resolution-ww-monthly-201002-201102>

StatOwl. 2011. [Verkkolähde] Web Browser Plugin Market Share [viitattu 15.1.2011]

Saatavissa: http://www.statowl.com/plugin_overview.php

Wikipedia Foundation, inc. 2011. [Verkkolähde] Adobe Flash [viitattu 5.1.2011]
Saatavissa: http://en.wikipedia.org/wiki/Adobe_Flash

Wikipedia Foundation, inc. 2011. [Verkkolähde] Canvas Element [viitattu 13.1.2011]
Saatavissa: http://en.wikipedia.org/wiki/Canvas_element

Wikipedia Foundation, inc. 2011. [Verkkolähde] Cascading Style Sheets [viitattu 6.1.2011]
Saatavissa: http://en.wikipedia.org/wiki/Cascading_Style_Sheets

Wikipedia Foundation, inc. 2011. [Verkkolähde] Google Maps [viitattu 20.1.2011]
Saatavissa: http://en.wikipedia.org/wiki/Google_Maps

Wikipedia Foundation, inc. 2010. [Verkkolähde] HTML [viitattu 15.12.2010]
Saatavissa: <http://en.wikipedia.org/wiki/Html>

Wikipedia Foundation, inc. 2011. [Verkkolähde] JavaScript [viitattu 6.1.2011]
Saatavissa: <http://en.wikipedia.org/wiki/Javascript>

Wikipedia Foundation, inc. 2011. [Verkkolähde] jQuery [viitattu 20.1.2011]
Saatavissa: <http://en.wikipedia.org/wiki/JQuery>

Wikipedia Foundation, inc. 2011. [Verkkolähde] Microsoft Silverlight [viitattu 6.1.2011]
Saatavissa: http://en.wikipedia.org/wiki/Microsoft_Silverlight

Wikipedia Foundation, inc. 2011. [Verkkolähde] Plug-in (computing) [viitattu 12.1.2011]
Saatavissa: [http://en.wikipedia.org/wiki/Plug-in_\(computing\)](http://en.wikipedia.org/wiki/Plug-in_(computing))

Wilder, R. 2000. [Verkkolähde] Learning to Use Color on Your Web Site [viitattu 2.2.2011]

Saatavissa: <http://www.creativepro.com/article/learning-to-use-color-on-your-web-site>